DOCUMENT RESUME

ED 205 170 IR 009 449

AUTHOR Towne, Douglas M.: Munro, Allen
TITLE Generalized Maintenance Trainer Simulator: Development of Hardware and Software. Final Report.
INSTITUTION University of Southern California, Los Angeles. Behavioral Technology Labs.
SPONS AGENCY Navy Personnel Research and Development Center, San Diego, Calif.
REPORT NO NPRDC-TR-81-9
PUB DATE Apr 81
CONTRACT N00123-78-C-1457
NOTE 29p.

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS Communications Satellites: *Computer Assisted Instruction: Computer Programs: *Microcomputers: Postsecondary Education: *Simulation: *Technical Education

ABSTRACT

A general purpose maintenance trainer, which has the potential to simulate a wide variety of electronic equipments without hardware changes or new computer programs, has been developed and field tested by the Navy. Based on a previous laboratory model, the Generalized Maintenance Trainer Simulator (GMTS) is a relatively low cost trainer that offers intensive training in equipment set-up, troubleshooting, and system assessment. It uses a software database and microfiche photographs to simulate electronic equipment, and the executive computer programs that manage all interactions between students and the system are programmed in UCSD Pascal. A database representing a satellite communications system was also developed to demonstrate the trainer's capabilities and to permit gathering of supportability data. The GMTS hardware and software have been refined and documented to meet the research objective of providing the intensive training desired at low cost. Testing and evaluation of the trainer in the school environment are scheduled. (MER)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED205170

U S DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

NPRDC TR 81-9

April 1981

GENERALIZED MAINTENANCE TRAINER SIMULATOR: DEVELOPMENT OF HARDWARE AND SOFTWARE

Douglas M. Towne
Allen Munro

Behavioral Technology Laboratories
University of Southern California
Redondo Beach, California 90277

Reviewed by
James S. McMichael

Released by
James F. Kelly, Jr.
Commanding Officer

Navy Personnel Research and Development Center
San Diego, California 92152

IR009449

FOREWORD

The research effort described herein was conducted in support of Project Z0789-PN, Class "A" Electronic Equipment Maintenance Training (EEMT) System, which is sponsored by the Deputy Chief of Naval Operations (OP-01). The objective was to demonstrate the concept of a general-purpose maintenance training simulator built from off-the-shelf hardware components and using software that was not limited to any single computer. Previous reports on this project described preliminary design options for a prototype EEMT system (NPRDC TN 79-3) and field tests of a stand-alone configuration of the BTL trainer (BTL Report 89 and NPRDC TR 80-30).

The prior contributions of the late Dr. Joseph W. Rigney of the University of Southern California in developing the concept for operational implementation are sincerely appreciated. Also, appreciation is extended to the following:

- Dr. Henry Halff, Office of Naval Research, for his review of an earlier version of this manuscript.
- Mr. Michael Andrews, San Diego State University Foundation, for preparing the software documentation in the Appendix.
- CWO3 Richard Merrell, Mr. Howard Quisenberry, and Dr. Norman Kerr, Chief of Naval Technical Training, for their continued support.
- The Director of the Advanced Electronic Schools Division of the Service Schools Command, San Diego, and his staff, particularly CWO4 Robert Doornbos, ETC Steven Gardner, and ET1 Kenneth Hutchinson.

The contracting officer's technical representatives were Dr. James S. McMichael, Mr. Russel M. Vorce, and Mr. George Lahey.

JAMES F. KELLY, JR.
Commanding Officer

JAMES J. REGAN
Technical Director

CONTENTS

	Page
INTRODUCTION	1
Problem	1
Background	1
System Description	1
System Development	2
Purpose	4
APPROACH	4
Training and Simulation Characteristics	4
Training Objectives	4
Training Exercises	5
Problem Identification and Selection	6
Student-trainer Interactions	7
Hardware	7
Software	10
System Software	10
Instructional and Simulation Software	10
Software for Data Base Development	12
Instructor Utility Programs	12
Simulation Data Base	12
Preparation	12
Organization and Structure	15
AN/WSC-3 Simulation	16
Operating Characteristics	16
Calibration	17
RESULTS	17
Operational Effectiveness	17
Support Requirements	18
Software Transportability	19
Instructor Utilities	19
CONCLUSIONS	19
RECOMMENDATIONS	19
REFERENCES	21
APPENDIX--SUMMARY COMPUTER PROGRAM DOCUMENTATION	A-0
DISTRIBUTION LIST	

LIST OF FIGURES

		Page
1.	General logic diagram of GMTS troubleshooting exercise	6
2.	GMTS training/simulation station	7
3.	Setting switches in GMTS	8
4.	Command menu.	9
5.	System software structure	11
6.	Hierarchical system representation	16

INTRODUCTION

Problem

As military hardware becomes increasingly complex and costly, the suitability of using the real equipment to meet all maintenance training requirements diminishes. Nevertheless, student technicians need to learn how the equipment normally responds to a wide range of conditions, and they need to experience abnormal responses and indications. As they develop a good understanding of equipment organization and functions, they need to develop and practice their skills in logical fault identification and isolation.

The limitations imposed by the real equipment in meeting these training needs are familiar ones. If the training command is fortunate enough to receive appropriate models of the real equipment, along with adequate spares, its maintenance problems are greatly aggravated by the use of the equipment in the training environment. As a training vehicle, the real equipment is limited by (1) the requirement that the instructor insert practice malfunctions, (2) the difficulty of capturing data regarding student performance, and (3) the cost of providing individualized assistance and guidance to each student.

Special-purpose simulators relieve some of these problems, but they too tend to be costly and difficult to maintain. Thus, the need exists for a general-purpose maintenance trainer that can be tailored via software to function as a trainer/simulator for a wide range of equipments and electronic systems. A laboratory model of a Generalized Maintenance Trainer Simulator (GMTS) has been developed and field tested (Rigney, Towne, Moran, & Mishler, 1980).

Background

System Description

The GMTS is a computer-based simulator that automatically selects malfunctions and displays high-resolution color images of the equipment. The student can rapidly access close-up views of any section of the equipment and can interact with the displayed switches by touching desired switch settings with a small stylus. Upon sensing such an action, GMTS determines what switch and setting have been touched, determines how the indications on the current image would be affected by that action, and automatically displays a new image. In a similar manner, a student may call up views of an equipment's test points, touch those of interest, and observe the indications that test equipment would produce. In effect, the student has the same opportunities for operating and troubleshooting the simulated equipment as he does with the real equipment. Ultimately, the student may identify the element that he suspects is faulty and "replace" it or he may give up. GMTS records all student performance for subsequent analysis and reporting.

The GMTS must not be confused with a trainer that attempts to represent a large class, or family, of equipments that share a common purpose. While such generic trainers may have great potential for instructing new technicians about an entire class of equipments, they nevertheless simulate the characteristics of one class. While the physical configuration of the GMTS is fixed, a subject matter expert (SME) can cause it in principle to simulate any target equipment or system. If desired, the GMTS can also serve as a generic trainer by simulating a real or hypothetical examplar of a family of equipments or systems.

Previous field tests of a laboratory model of GMTS demonstrated the potential of the concept, but the laboratory model was limited in the following ways:

1. High-level programming languages were unavailable, so the executive programs were written in assembly language. The software was thus committed to a particular microprocessor, and programs were not easily expanded and revised.

2. Standard peripheral interfaces were unavailable, so they were custom engineered. The reliability and maintainability were poor, and the unit could not be replicated at a low cost.

3. The commercial subsystems comprising the trainer/simulator became essentially obsolete during the final development year. Smaller, cheaper, and more reliable counterparts became available.

4. Many of the functions that instructors would typically perform in the instructional environment were performed by the contractor. Additional documentation and utility programs were required if the trainer was to be turned over to a training facility.

System Development

The basic concept of GMTS emerged after many years of efforts to represent complex physical systems and human behaviors in a form conducive to machine processing. Whereas most research in computer-aided instruction derived from questions about the nature of learning and the structure of knowledge, this early work was concerned solely with characterizing the operation of a physical system. The result was an explicit and general taxonomy of the nature of normal and malfunctioning systems. These representations, in the form of computer programs, expressed the general processes, both manual and cognitive, performed by a maintenance technician, and the types of states in which systems can exist. The necessary inputs to these programs represented the data necessary to characterize any particular system.

Through the mid 1960s, this research was concerned with modeling corrective maintenance performance and predicting associated repair times. Such modeling required (1) a model of an "ideal" troubleshooter, (2) a theory of human troubleshooting behavior, and (3) a general-purpose representation of the system to be maintained. These requirements are described in the following paragraphs:

1. A model of an ideal troubleshooter. This model was based upon a Bayesian fault isolation process (Rigney, Cremer, Towne, & Bond, 1967). In this process, subjective probabilities are associated with each element in a set of hypotheses. The initial probabilities of the N elements may either be based upon available reliability data or all may be set to 1/N.

The Bayesian model selects as the next test the one that is expected to yield the maximum uncertainty reduction, which is a function of the a priori probabilities and the information value of each test.

A second computer program (Towne, 1968; Towne & Mason, 1967) was developed to generate the time required by a human technician to isolate faults, if he followed the Bayesian algorithm. This program automated the application of classical industrial engineering techniques to produce a time "standard" for a defined work method. A limitation of that technique was that only manual time could be projected. At present, this approach is being generalized to include cognitive time.

2. A theory of human troubleshooting behavior. Using the Bayesian process as a baseline, which, incidentally, reduces the "half-split" technique under certain conditions,

an effort was made to ascertain the extent of conformance between this model and human strategies. Specifically, we investigated (a) whether technicians with better data (i.e., better understanding of the system in question) were more Bayesian, and (b) whether they would appear more Bayesian if their data (rather than the true symptom-malfunction (SM) data) were used to drive the model.

Approximately one-half of the technicians studied (N = 39) were found to be generally Bayesian when using their own SM data; the other half were not. However, even the "Bayesian" technicians were only about 30 percent as efficient as the Bayes model using true SM data.

The correlation between the accuracy of subjects' SM data and their tendency toward Bayesian troubleshooting was a moderate r = 0.55. Whenever subjects erred in generating SM data, they generally made several more irrational non-Bayesian actions.

3. A general-purpose system representation. The research described above motivated an effort to develop a generalized and machine-computable technique for representing a real system. Consequently, a program was developed (Rigney & Towne, 1972, 1977) that accepted a straightforward itemization of system elements including general-purpose blocks, switches, relays, switch wafers, signals, indicators (including test points), and modes.

This system was applied to the AN/SPA-66 radar repeater, producing a system description involving 38 network blocks, 51 indicators, and 17 mode switches. The program was able to generate correct SM data in any mode of operation. These SM data were subsequently processed to produce Bayesian troubleshooting strategies and to evaluate human-generated strategies. This program was then employed to generate student-computer dialogue through a teletypewriter connected by telephone to a time-shared computer (Rigney & Towne, 1974). The computer-generated statements were produced by combining fixed statement patterns with text from the equipment-specific data base. An example statement is shown below:

YOU SHOULD KNOW THAT <system element name> CANNOT BE FAILED

BECAUSE YOU OBTAINED <symptoms> WHEN YOU PERFORMED <test name>.

A relatively wide variety of statement forms yielded the ability to make suggestions, to correct conceptual errors, and to respond to various calls for assistance.

By the early 1970s, minicomputers were available that were capable of executing the GMTS program and controlling other peripherals. In 1978, the first informal field test of the GMTS stand-alone configuration was conducted when the trainer was applied to the Fleet Communications System, a large multi-equipment system for radio communication (Rigney, Towne, King, & Moran, 1978). In that test, 20 class "A" school students worked on 35 simulated troubleshooting problems. Results indicated that the trainer offered realistic, accurate, and efficient practice and that the learning transferred well to the real equipment.

In a second field test, the GMTS was applied to an entirely different target system than that used in the initial test; namely, the AN/SPA-66 radar repeater that had been simulated in the early research (Rigney, Towne, Moran, & Mishler, 1980). The data base for this test was prepared by two technical experts who were concerned only with supplying the specified data in the required formats. A relatively short field test followed, involving 10 subjects, each attempting to isolate 33 simulated malfunctions over

a 16-hour period. Following this practice phase, the students were tested using an actual AN/SPA-66 with actual inserted malfunctions. As with the first test, results were generally positive, especially in relation to success in the test phase using actual equipment. Because of the small sample size, however, the primary value of this test was to demonstrate that technical content experts can apply GMTS.

Having passed those early tests, the time had come to compare the training effectiveness of the GMTS to that of real equipment. First, however, the laboratory model had to be upgraded, reprogrammed, and extended to allow a realistic implementation in the training environment.

Purpose

The purpose of the research described here was to improve the laboratory model, maximize the transportability of the software to new microprocessors, and augment the software to allow Navy instructors to manage the system.

The specific objectives for improving operational effectiveness were to produce a smaller, cheaper, and more reliable configuration that could be easily replicated and maintained by others. The ease of implementing the software on new microprocessors would be maximized by reprogramming the GMTS executive program in the high-level language, UCSD Pascal ("UCSD Pascal" is a trademark of the Regents of the University of California), and by producing a complete software documentation package.

Finally, a number of extensions to the existing repertoire of instructional facilities were specified. The primary thrust of these extensions was to increase the extent to which the instructor could shape the student-trainer interactions and retrieve student performance data.

APPROACH

Training and Simulation Characteristics

Training Objectives

The specific training objectives to be achieved using GMTS may vary greatly from one application to another. The general objectives for which the trainer was designed are as follows:

1. To enable the student to become proficient in determining the state of the real equipment being simulated; that is, whether the equipment is operating normally, operating within the tolerances and limits of fully operational units, or is not fully operational. For equipments not fully operational, the student will be able to identify normal and abnormal operating modes to determine the extent to which the equipment is degraded.

2. To enable the student to become proficient in setting up and interpreting front panel tests. This includes establishing desired modes of operation and interpreting the normality/abnormality of indications exhibited in these modes.

3. To enable the student to become proficient in the selection and use of test equipment. Proficiency should include identifying and attaining prime equipment modes that effectively exercise the functions being considered, selecting test points that will

reveal the states of the prime equipment in these modes, and interpreting the indications obtained.

Simulations of prime equipments will usually not include full simulation of general-purpose test equipment. (Such exercises could, however, be provided by the trainer in the future.) The training objectives related to use of test equipment focus on choice of test points to be sampled, choice of prime equipment modes during that sampling, and interpretations of test results.

Training Exercises

The training exercise explicitly addressed by GMTS presents the student with an initial statement of a malfunction's gross effect(s) and requires that he employ trouble-shooting skills to isolate the source of the problem. A logic diagram of that process is shown in Figure 1.

GMTS can also provide other types of part-task exercises, such as the following:

1. Equipment familiarity. Following written guides, the new technician can set up the equipment in major operational and maintenance modes to become acquainted with normal system operation. This exercise closely resembles an exercise commonly conducted with the real equipment in many military electronics schools.

2. Set-up exercise. Working without aid of detailed instructions, the student can practice achieving modes named by the instructor. If desired, the instructor can provide some selected readings at various test points, thereby allowing the student to determine if his set-up is correct. Alternatively, as a test, the student can turn in his readings at selected test points.

3. Test point location exercise. The student can locate and sample the values of various test points designated by the instructor. This exercise would provide practice in locating test points by their designations in the technical manual.

4. Symptom evaluation exercises. Under various malfunction conditions established by the trainer (under instructor control), the student can extract symptoms information from various indicators and test points in designated modes. The exercise consists of assessing the normality of each observed indication.

5. Functional assessment exercises. This exercise is concerned with establishing relationships between observed indications and possible causes of these data. In one form, the student simply proceeds from exercise four, above, to itemize the possible causes of the observed abnormal indications. The instructor could then evaluate these lists individually, or in class. In an alternate form, the student predicts the responses of various indicators in known malfunction conditions after studying the technical documentation. He then operates the trainer to determine if his predictions are valid.

These simple exercises can be produced by an instructor using the standard instructor utility functions that give control over selection of the malfunction, if any, and order of problems.

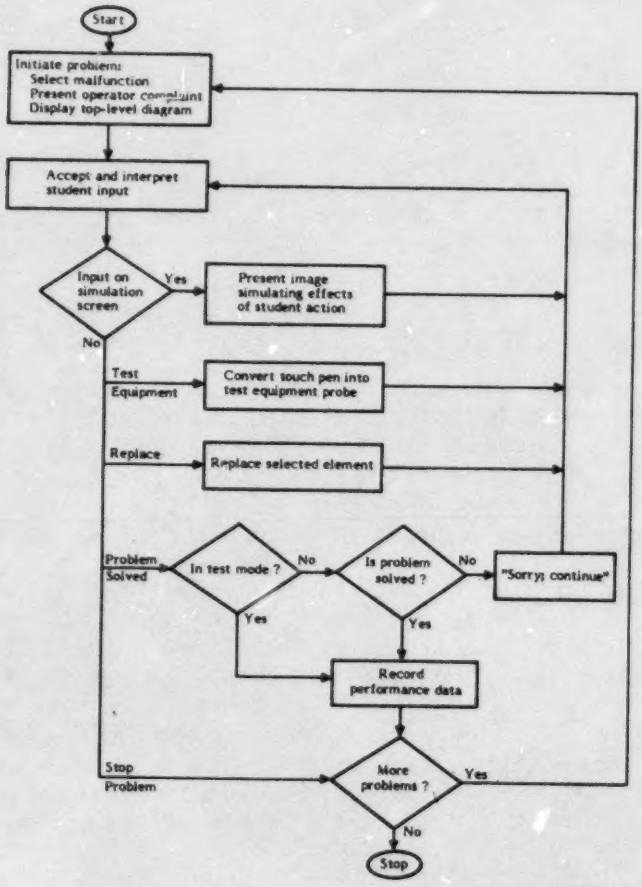

Figure 1. General logic diagram of GMTS troubleshooting exercise.

Problem Identification and Selection

The repertoire of malfunctions to be presented by the trainer is determined by the data base preparer. GMTS can simulate any malfunction that exhibits its symptoms in a consistent fashion (intermittent faults are not permissible). A cascading fault, which in the real equipment would cause one or more other faults, is permissible. The set of resultant faults, however, along with the causing fault, must all be described as a single "malfunction syndrome" by the data base preparer, and must all be replaced at one time by the student. That is, the system will not simulate the temporary correction and subsequent recurrence of resultant symptoms if the causing failure persists.

The executive program determines the next problem to be presented for each student according to instructor-supplied specifications. During data base development, the instructor supplies the practice and test problems to be presented and allocates these problems into sets called pools. At one extreme, one problem could be placed in each pool, achieving a fixed sequence of presentation. At the other extreme, all problems could be placed in one pool, achieving complete random sequence of presentation. More commonly, instructors assign problems to pools that relate to planned lectures.

At any time during a course, the instructor may key in the highest pool number from which problems may be selected, thus assuring that students receive only problems that are appropriate to the progress of the lecture series.

Student-trainer Interactions

Conventional computer-assisted instruction typically proceeds through cycles of content presentation, question presentation, answer processing, and subject-matter routing. GMTS currently relies on external means, such as lectures and text books, to present content such as theory of equipment operation. The trainer then provides the student with a means for (1) exploring a complex system by operating it in its normal state and a variety of abnormal states and (2) developing and refining a logical fault isolation process as a function of increased understanding of symptom information related to modes and malfunctions.

Hardware

The GMTS (Figure 2) consists of three off-the-shelf commercial units:

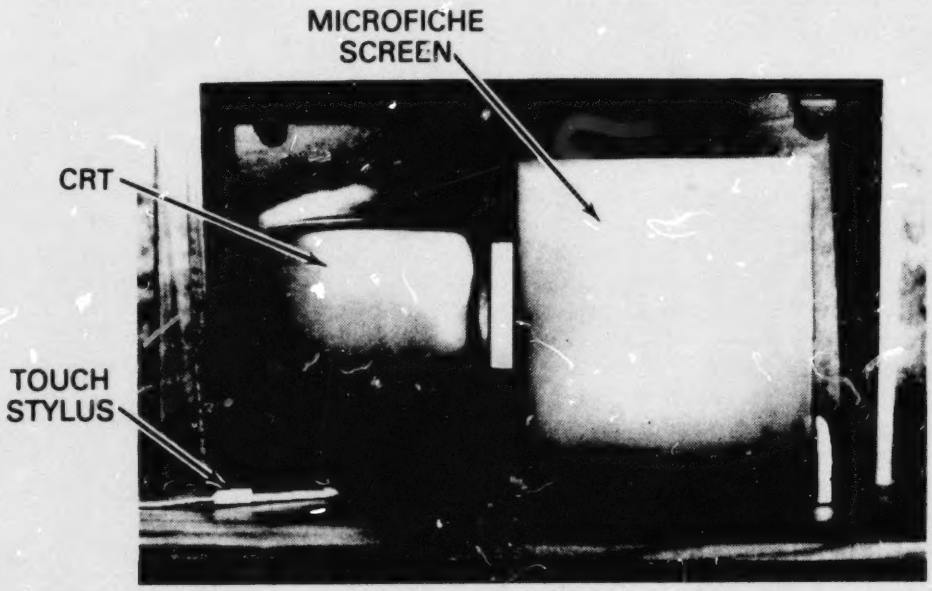

Figure 2. GMTS training/simulation station.

1. A Terak 8510a computer, with 28K words of 16-bit RAM, a real-time clock, a graphics and alphanumeric CRT, an 8-inch floppy disk drive and keyboard (used only by the instructor), and a second floppy disk drive (Terak 8512).

2. A Bruning Model 95 microfiche projector with an RS232-C computer interface and 1,800-image capacity (30 fiche x 60 images per fiche).

3. A Science Accessories Corporation Grafpen with an RS232-C computer interface and 1mm resolution.

The three major units are plug compatible; that is, they are connected without electronic modification via standard cables. The only hardware modifications were to remove the manual keyboard from the Bruning 95, the two Grafpen sonic sensors from the standard mounting, and the Terak CRT from its standard cabinet. The units were then installed in a modified Emcor Data Desk, which provided a lockable bay for disk drives and other electronics, ample working space for handling and studying technical manuals, and an integrated, enclosed unit.

This configuration provided two displays: (1) the computer-driven CRT used primarily to interact with the student regarding instructional and tutorial matters and (2) a rear-projected microfiche screen for representing the real equipment.

The Grafpen provides the means of input to the trainer, and all student actions are made by touching the CRT screen, the microfiche screen, or the command menu with a stylus whose position can be sensed by the computer to within 1mm accuracy. The CRT acts exclusively in an administrative, vice simulation, capacity. It displays statements and questions to (1) identify a student at the start of a session, (2) allow the instructor to designate the latest lecture completed, (3) initiate new problems, (4) execute replacements requested by the student, (5) verify current test equipment selections and connections, and (6) handle termination of problems and sessions. The microfiche screen acts exclusively to represent the real equipment.

At the initiation of a new problem, the CRT displays an operator's complaint, such as:

THE MALFUNCTION LIGHT IS ON,

which is similar to a failure report initiating a real repair effort. Simultaneously, the microfiche screen displays a "top level" diagram that exhibits all equipments and peripheral systems that the student can manipulate or examine. The student obtains closer and closer views of the equipment by touching the area of interest with the acoustic stylus. This incremental "zooming" normally involves one to three steps, which result in a close view of a relatively small portion of the entire system. Whenever a close-up image is shown, front-panel indicators will appear identical to the ones on the real equipment, given the mode of operation and failure condition, if any.

The student changes switch settings by touching the panel, as shown in Figure 3. Whenever the student touches a new switch setting or test point with the stylus, the computer commands the microfiche projector to retrieve and display the image that reflects the action taken.

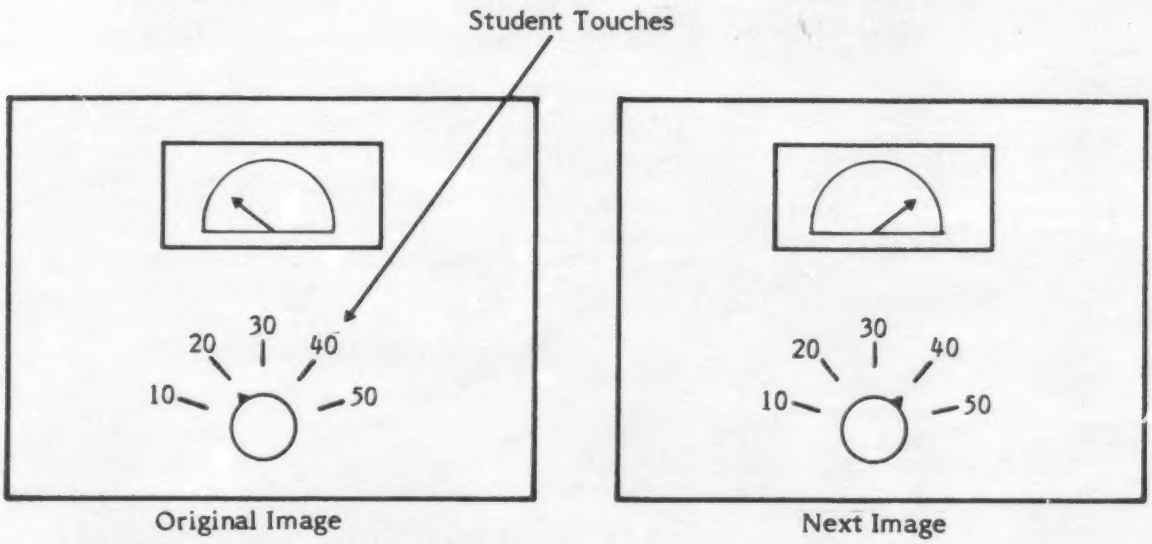

Figure 3. Setting switches in GMTS.

The command menu, shown in Figure 4, allows the student to interact with the executive program.

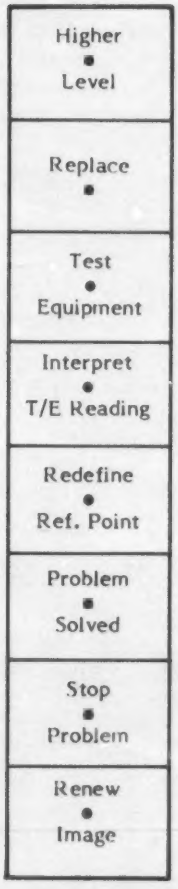

Figure 4. Command menu.

The commands and their effects are as follows:

1. HIGHER LEVEL. Students touch HIGHER LEVEL if they want to see the next less detailed picture in the system's hierarchical structure. This was the picture from which the currently projected section of the system was selected.

2. REPLACE. Students touch REPLACE if they want to replace the section of the system currently displayed. Subsequent indications will reflect the success of this action.

3. TEST EQUIPMENT. Students touch TEST EQUIPMENT if they want to perform test equipment reading(s). The CRT displays a list of all test equipment defined, and the test points to which they are currently connected (or "-----" if not connected). For example, the following display shows that a multimeter is connected to test point J14 on card 1A1A2:

TEST EQUIPMENT:	CONNECTED TO:
*OSCILLOSCOPE	-----
*MULTIMETER	1A1A2 J14

Touching the asterisk (*) next to the desired test equipment name in effect causes the touch stylus to become the probe for that type of test equipment. Touching a displayed test point produces the test-equipment reading on the microfiche screen and the test point designation on the CRT.

4. INTERPRET T/E READING. Students touch this command if they want help in determining whether the test equipment reading currently displayed on the microfiche screen is normal or abnormal. This support is unavailable in test mode, and its use is recorded with the student's performance data.

5. REDEFINE REFERENCE POINT. Students touch this command if they want to repeat the calibration of the currently displayed image by touching a special symbol in the corner of the image.

6. PROBLEM SOLVED. Students touch this command if they believe they have restored the simulated equipment to normal operation. In practice mode, students are allowed to continue a problem that has not, in fact, been solved. In test mode, this command terminates a problem.

7. STOP PROBLEM. This commands the trainer to abort the current problem.

8. RENEW IMAGE. Touching RENEW IMAGE causes the microfiche projector to reproject the current image in case of poor registration.

Software

System Software

The system software described in this section consisted of computer programs written in UCSD Pascal. Particular advantages of developing programs in UCSD Pascal are listed below:

1. Pascal is highly standardized, maximizing ease of augmenting and maintaining programs.

2. Pascal was specifically developed for ease of program transportability (i.e., implementation on other computers).

3. The structured nature of the language tends to promote well organized programs that are easier to develop and document.

4. The UCSD Pascal operating system offers a complete filing and editing facility, which is used extensively during data base entry.

Instructional and Simulation Software

The computer programs that control the (on-line) delivery of instruction and simulation are of two types:

1. Simulation programs that sense and interpret student actions, determine the effects of those actions in the real equipment, and respond by altering the presentation of the equipment on the simulation screen. The major components of simulation control are listed below:

a. Student action interpretation--Sensing and interpreting the significance of an action by the student, as indicated by a touch-pen strike.

b. State evaluation--Determining the state of an equipment (including test equipment) from data describing a malfunction and the current mode.

c. Selecting the appropriate microfiche display.

2. Pedagogical programs that select problems, provide symptom assessment assistance (INTERPRET T/E READING), record performance data, and offer other instructional functions.

Figure 5 illustrates the general organization of the GMTS system software, and the appendix provides summary program documentation.

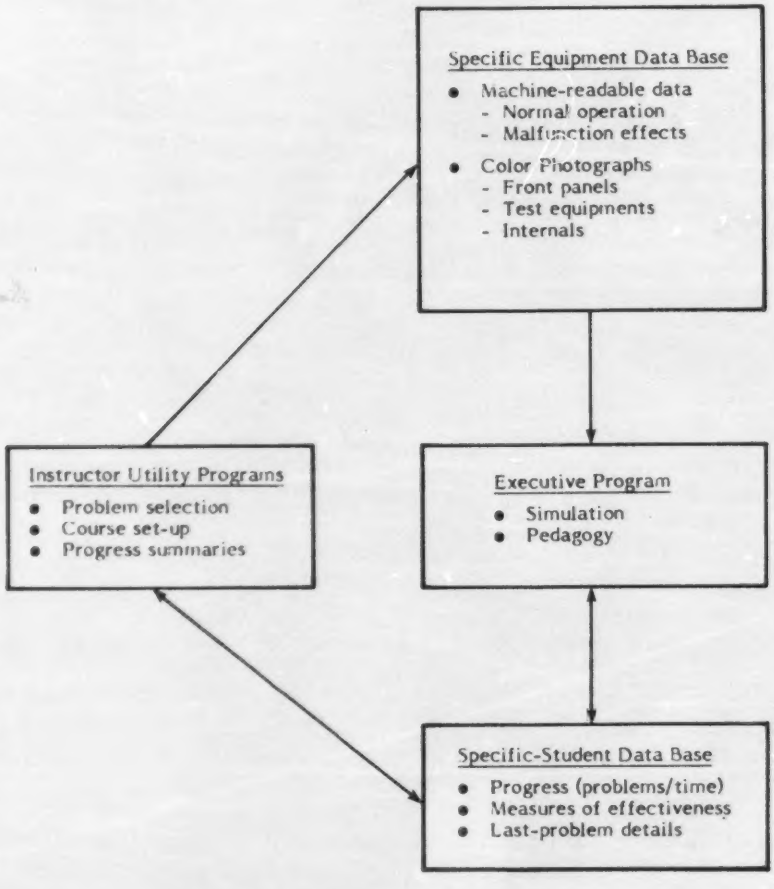

Figure 5. System software structure.

Software for Data Base Development

After organizing the alphanumeric data to describe the equipment, the SME keys the data in at a keyboard that plugs into the computer. The resulting data file may subsequently be printed and edited, using system software provided with the system.

When this data file is prepared, a utility program is executed that reads in the source data, converts it to a format that can be more rapidly accessed by the training program, and checks for a number of possible data errors. The resulting file is then copied to diskettes for use by students. If desired, the data base preparer may execute a second utility program to verify the data base. This program operates on the data base exactly as does the training program, except that it accepts keyboard input only and generates CRT output only. Its purpose is to allow the data base preparer to run the simulator prior to producing the microfiche.

The third utility program used to support data base development is concerned with defining the various points on each microfiche image. This program displays each microfiche image, and the SME can designate subpictures, test points, and switch positions with the touch pen. As this is done, he identifies various points by touching key words and numbers displayed on the CRT. At the completion of this process, a data file is automatically produced to allow the trainer/simulator to interpret each student input.

Instructor Utility Programs

Instructor utility programs are provided to establish problem selection constraints, set up new classes, and produce class progress summaries. The special-purpose utilities available to instructors are as follows:

1. ADD STUDENTS. This utility is used to prepare student diskettes for new student data. It reserves space on each diskette for student data and heads that space with the new student's name.

2. GET STUDENT DATA. This utility copies student performance data from each student's diskette onto a class master diskette.

3. PRINT STUDENT DATA. This utility prints out a record of each student's performance data in the class to date, providing instructors with data on individual student performance.

4. SET PROBLEM SEQUENCE. This utility allows instructors to establish a newly required sequence of problems.

5. SEE PROBLEM SEQUENCE. This utility allows instructors to review the prescribed sequences of individual problems or problem tools.

Simulation Data Base

Preparation

The simulation consists of (1) the alphanumeric data base required to compute responses of the equipment to student actions and (2) color microfiche to simulate the appearance of real equipment. The alphanumeric data include the following:

1. Names of replaceable elements.

2. Operator's failure complaint (to start each problem).

3. Names of test equipments available.

4. Names of test points (for verification on the CRT).

5. Lists of allowable test equipments for each test point.

6. Names of switches and settings.

7. Initial settings of each switch (by problem if desired).

8. Modes of operation.

9. Symptom-malfunction relationships by mode.

10. Required equipment mode for replacement (usually deenergized).

11. Index to microfiche images.

In addition, a "map" of each section of the equipment is created by running the utility program XY (see appendix). This map is a list of Cartesian coordinates that identifies each active point (i.e., one which may be touched by the student with the input stylus).

With this rich base of textual and logical data, GMTS is able to do the following:

1. Select an appropriate malfunction in accordance with the instructor's plan and the student's progress.

2. Retrieve and display the operator's complaint to initiate the problem.

3. Project the "top level" diagram of the system.

4. Display successively closer views of the area of the real equipment indicated by the student.

5. At the close-up level where symptom information becomes visible, compute the symptoms evident at each indicator in the currently viewed section of equipment and project the image containing these symptoms.

6. Respond to touch inputs at switch settings by recomputing all symptom data and projecting the image reflecting both the switch setting change and the new symptoms, if any.

7. Respond to requests for replacement by subsequently producing normal indications, if the replacement was correct.

8. Respond to requests for symptom interpretation assistance, if not in the test mode.

9. Record student performance data, including errors and time.

While this data base is necessarily complex and somewhat voluminous, an SME can learn to develop such a file in 1 or 2 weeks. The only knowledge necessary is a thorough understanding of the equipment to be simulated.

A complete guide has been prepared that describes the data files in detail, as well as the process for developing them. This document will be published separately as a GMTS User's Manual. The steps used in assembling the information are described below:

1. Define scope of the training simulation (system analysis). The scope of the simulation is determined by the modes in which that hardware is to be simulated. The modes are defined in a listing of all prime equipment and test equipment, including any peripheral equipment that will be either operated or observed by the student as he troubleshoots.

2. Analyze training requirements. This step consists of determining the level to which maintenance is performed (i.e., will modules, boards, or components be replaced?), defining the prerequisite skills of entering students, defining terminal training objectives, and, finally, selecting malfunctions for simulation that will best offer the practice necessary to attain those objectives.

Careful consideration should be given to student skills necessary for troubleshooting and to the maintenance philosophy of the equipment. These impact the depth of knowledge the student must possess to repair the equipment and the choice of effective practice problems. The two general types of problems are those selected for their instructional value in illustrating a function of the unit, and those selected as a result of their incidence in the fleet.

3. Define the equipment states to be simulated. The elements defined by Step 1 form a first level of subdivision of the entire system or equipment. Each of these elements must be further subdivided. Nearly any system can be viewed as consisting of no more than six to ten elements. Furthermore, even large, complex, multi-equipment systems rarely require more than three to four levels of division to reach meaningful groupings of dials, indicators, or test points. This means that the user will normally be able to access an area of current interest with only one or two changes of level, each accomplished by touching the stylus to the area of interest on the projected image.

Within each area of interest, the data base preparer must decide how many discrete positions of continuous controls and continuous indicators (such as meters) will be simulated. Generally, three to five positions are adequate. At this step, the data base preparer enumerates the number of images required to represent the modes of operation defined in step one. If the number is excessive (over 1,800 using current micrographics hardware), one must either reduce the number of modes to be simulated or divide the equipment into smaller sections.

4. Photograph the real equipment in each state and produce color microfiche. Two types of pictures are required to support the simulation: (a) pictures of equipments or portions of equipments, which serve as pathways to more detailed subpictures, and (b) pictures of all the states defined for sections with controls and/or indicators. The former, termed intermediate scenes, are often produced via a combination of photographs and graphics art. The latter, termed multistate scenes, are photographed using a detailed picture list as a guide. The picture list defines each possible state (both normal and abnormal) of each multistate scene, in terms of the control positions and indications required. Color 35mm slides are then taken of each equipment state and sent in for commercial microfiche production.

5. Encode the list of states, index to microfiche images, and symptomatic function data. In general, each simulated problem must deal with (a) equipment malfunctioning due to a fault, and (b) equipment now functioning properly, since the student has replaced the faulty element.

The data base preparer first collects the data required to simulate the normal operating equipment. For each section of the system, the normal pictures are noted and the switch settings required to produce each picture are listed. Next, the data base preparer specifies the effects of each malfunction by identifying the sections of the system, including test equipment, that are affected by the malfunction.

The amount of data listed is greatly limited since (a) only exceptions to normality are identified and (b) for any one section, many malfunctions will cause identical abnormal symptoms. This commonality is exploited to minimize the data required to characterize the effects of a malfunction.

In addition to the data described above, the preparer also inputs such information as initial switch settings and test point constraints (i.e., what test equipment can be used on various test points?).

6. Key in the data and check the resulting file. The information compiled in Step 5 is keyed in at a trainer station and recorded on a diskette. Then a program is executed that examines the data, checks for omissions and contradictions, and reformats the data in machine-readable format. After any noted errors are corrected, a second utility program is executed to identify the locations of all points that the user may touch with the touch pen. This is accomplished by touching each "active" point in the picture and then providing the associated subscene number, switch number, setting number, or test point number.

The data base structure is flexible, and it may be desired to carry the fidelity and detail much further in some areas than others. Also, there are often many opportunities for greatly paring the number of system states simply by eliminating redundant modes, bands, etc. Such reduction can often be done without artificially simplifying the troubleshooting task. For example, if a transmitter operates in 32 bands, the simulation may only need to implement two or three.

Organization and Structure

A functional representation of the graphic portion of the simulation data base is presented in Figure 6. While this figure happens to represent a system with three levels, there may be up to ten levels of hierarchy employed. The system software does not require a uniform decomposition of subelements. Thus, some equipments in a system may require four or five levels and others, only one or two. Finally, the system software imposes no constraints on the contents or layout of each microfiche. Thus, most rapid response can be achieved by grouping together all images for a particular equipment section.

The alphanumeric data base is written to a single 8-inch diskette, along with a copy of the system software. Thus, the trainer can be changed to simulate a different equipment by changing the microfiche cassette and the data base diskette.

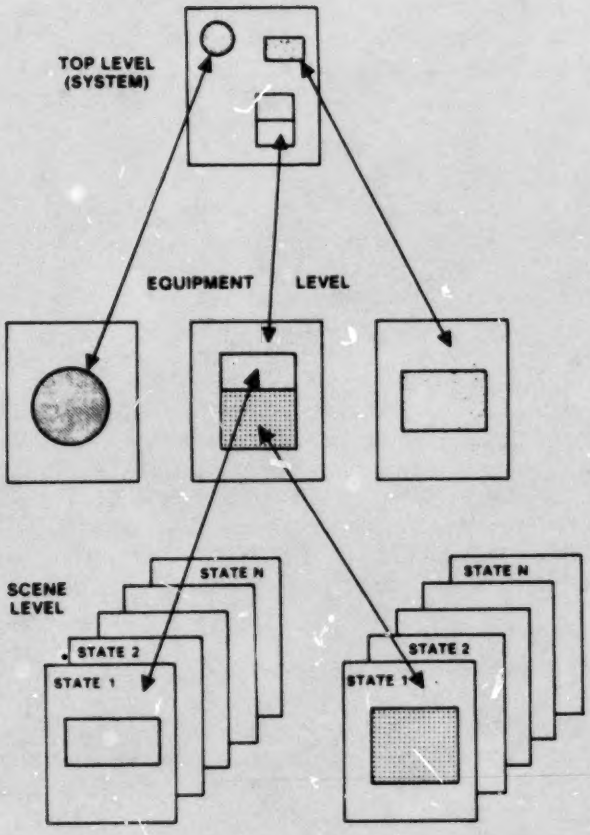

Figure 6. Hierarchical system representation.

AN/WSC-3 Simulation

The AN/WSC-3 is a transceiver that is the major component of a fleet satellite communication (SATCOMM) system. The SATCOMM system includes, beside the AN/WSC-3, two antennas, amplifiers, relays, a teletype, and several minor components. The simulation data base was developed for the entire system, and the largest part of it related to the AN/WSC-3 itself.

Advanced skills training for the maintenance of the SATCOMM system is provided by the Advanced Electronic Schools Division (AESD) of Service School Command, San Diego. This school was the testbed for developing the data base and for gathering data on the supportability of the GMTS. Six advanced development GMTS units were constructed, four of which were installed in the AESD school for data-base development and for experimental use by students. The results of these two efforts will be the subject of a future report.

Operating Characteristics

GMTS response time to touch inputs varies according to touch location. Response to touching the CRT or command menu occurs within 0.1 second. The trainer responds by altering the CRT display, updating the simulation display, or, if the touch was not identifiable, emitting an audible beep.

Responses to inputs on the microfiche screen involve a compute delay plus image change time. The compute delay, which varies from one data base to another, ranges from less than 0.001 second to approximately 5 seconds, with an average of approximately 1 second. Image change time varies from approximately 1 second, if the new image is on the same fiche as the current image, to approximately 3 seconds, if a fiche change is required.

Calibration

Although the touch pen inputs are detected to 1mm precision, the student is required only to touch the pen within a reasonable proximity to the desired point. GMTS interprets touch inputs by comparing the detected touch location to a set of defined, stored points. This process, however, requires a means for correcting for the following random hardware variations:

1. Variation in location of the CRT, touch-pen sensors, and command menu (from one training unit to the next).

2. Day-to-day variation of display location on the CRT.

3. Day-to-day variation in touch-pen operation.

These variations are measured each time a unit is energized. The calibration consists of touching displayed targets on the CRT and command menu, thereby identifying their exact location. This also verifies that the touch-pen and CRT are functioning correctly.

A more bothersome variation is the positional error in projecting microfiche images. As a result of the high magnification employed (22X), very small mechanical deviations result in errors as large as 0.5 inches in absolute position of an image on the screen. Normally, this error would not cause difficulty; however, some images may contain test points or switch settings that appear less than 1 inch apart and could be confounded. To overcome this variation, each microfiche image contains a small "bulls-eye," called a reference dot, in the lower right-hand corner. Each time a new image is displayed, the user first touches the reference dot. The GMTS then corrects all subsequent user inputs on that image by the calculated error offsets in the horizontal and vertical dimensions. If micrographics projectors now under development can deliver repeatable image position with errors less than 1/16 inch, there will be no need for the reference dot.

The error correcting performed by the executive program allows the positional data stored on diskette to be independent of the particular unit that will employ it. Furthermore, it allows free substitution of subsystems (touch pen, microfiche projector, CRT) without upsetting the reliability with which user inputs are interpreted.

RESULTS

Operational Effectiveness

The upgraded configuration of GMTS is smaller, lighter, and more economical than the laboratory model. While some aspects of user acceptance and training effectiveness can be determined only after more extensive experience in the training setting, it appears that the system's response time, reliability, maintainability, and quality of simulation are very satisfactory. Specific findings are listed below:

1. The GMTS response time averaged 1 to 2 seconds. The worst cases, 5 to 8 seconds, occurred in less than 5 percent of the system responses.

2. Student acceptance of the touch pen was high. Inputs are definite, as they are accompanied by a faint click, and recognition by the GMTS was excellent.

3. Image quality on the microfiche screen was sensitive to the wear of the plastic fiche holders covering each fiche. As the holders became scratched and opaque from wear, the quality of the image deteriorated markedly. While the fiche itself is not affected, the holders must be replaced periodically.

4. Given proper preventive maintenance attention, the microfiche retrieval system functioned quietly and smoothly. On occasion, the unit either failed to retrieve and project an image or projected a double image. On the average, this occurred approximately once for every 2 or 3 hours of use. When this happened, the student could touch RENEW IMAGE at the command menu to correct the image projection. There was no instance in which an incorrect image was displayed.

Support Requirements

Table 1 presents projections of subsystem reliability and maintainability for the GMTS. The per unit failure rates, maintenance times, and repair costs are based upon 18 months' experience with six units in a developmental environment and 4 months' experience with four units in a training setting.

Table 1

System Reliability and Maintainability

Unit (Subsystem)	Projected Failures Per Year	Average Man-hours Per Failure	Preventive Maintenance Hours/Year	Total Man-hours Per Year	Projected Parts Costs ($) Per Year
Terack System	1	8	20	28	200
Grafpen	1	10	10	20	100
Bruning 95	10	1	50	60	200
Printer	1	4	10	14	50
System (Cables, etc.)	2	1	10	12	10
Total				134	560

Reliability of the computer, CRT, disk drives, and touch pen has been excellent. As shown in Table 1, only two failures per year are projected for these components. Restoration times can average 8 to 10 hours for these failures, assuming that malfunctions will most likely occur in the mechanical portion of the disk drive or will be intermittent. Hard failures on circuit boards, however, could be corrected in less than 2 hours.

The most complex electromechanical device in GMTS is the microfiche retrieval unit, which is projected to fail ten times per year. Restoration time is expected to be short, however, as most failures are likely to involve jamming or slippage.

The 8-inch floppy diskettes have proven to be economical and reasonably durable. While extreme abuse will destroy the stored data, instructors can easily copy daily data to master diskettes and print hard-copy summaries. Thus, catastrophic losses of student data can be prevented.

Software Transportability

The executive and utility programs were fully reprogrammed in UCSD Pascal. No system function required assembly language patches or nonstandard application of the language. These source programs have been compiled under the I.5, II.0, and II.1 versions of the Pascal compiler for the Terak 8510a, the Apple II, and the Altos ASC8000. Since UCSD Pascal can be implemented by an increasing number of microcomputers, GMTS software can capitalize upon future hardware improvements. The only machine-dependent software employed was related to interfacing with the touch input system and the microfiche system.

Instructor Utilities

The repertoire of utility programs for instructor use was expanded to include programs to set up diskettes for new students, to control problem presentation, and to copy, accumulate, and print out student performance data. In addition, utility programs were produced for use during data base development. These programs check and reformat the raw input data and provide a convenient means for identifying touch points on the graphic images.

One course has been conducted to train SMEs in data base preparation. This experience indicated that SMEs can be trained in less than 2 weeks to prepare a new data base and execute the associated utility programs.

CONCLUSIONS

The GMTS hardware and software have been refined and documented to meet the objectives of this phase of research. The trainer is ready for test and evaluation in a Navy school.

UCSD Pascal is a suitable language for GMTS programming, and such programs are transportable to other computers.

RECOMMENDATIONS

The software products that are described in this report are potentially useful to the Navy beyond this research. The UCSD Pascal operating system and language should be used, along with the GMTS computer programs, for the prototype electronic equipment maintenance training (EEMT) system (Device 11B106), which is being developed for initial-skills training of electronic technicians and electronic warfare operators. While the data base that simulates the AN/WSC-3 was developed for advanced-skills training, it could be modified for use by the EEMT project. Different malfunctions would have to be simulated, but the normal operations data base could be used intact. The microfiche

photographs would have to be replaced with media that are compatible with Device 11B106, which uses videodisc.

Numerous questions regarding the feasibility of various pedagogical approaches and design alternatives for general-purpose maintenance simulators remain to be addressed by future research:

1. Training effectiveness. Before the training effectiveness of general-purpose maintenance trainees can be determined, the following pedagogical issues should be addressed:

   a. The amount of student control that is optimal in various situations.

   b. The types of support and aiding that should be provided.

   c. How effectively an intelligent trainer can adapt to particular student needs (beyond pace of training and problem selection).

Many of these pedagogical issues are entwined with design variables. For example, we need to know the relative advantages and disadvantages of a single display trainer and how data base preparation cost is affected by computer-generated graphics.

2. Fidelity. Research should be conducted to determine the need for three-dimensional maintenance simulators, which are realistic physical simulations of the real equipment. At a middle point on the realism continuum are hybrid trainers, which offer hands-on use of actual test equipment in association with "flat-panel" or two-dimensional simulation of the less universal real equipment. Such combinations may provide a very attractive combination of cognitive exercise and procedural drill and practice.

3. Representation. The process of preparing data bases for GMTS should be streamlined. Since much of that process is clerical, a computer-aided approach may yield considerable savings.

4. Application. We should learn more about the domain of the GMTS approach. The only assumption made about the inherent nature of the simulated equipment is that it can exist in a number of discrete states. Experience with applying GMTS to three large electronic systems indicates that it is sufficiently general to accommodate nearly all electric and electronic simulations. Those functions that do not strictly meet the assumptions can almost always be reduced or restricted in ways that impose minimal artificiality. For example, a continuously-reading meter can be defined as indicating a small number of values, such as 0, low, normal, high, and maximum.

One question that remains is the extent to which nonelectronic systems could be accommodated; that is, whether automatic boilers, engines, antenna mounts, and helicopter control mechanisms could be represented adequately by this finite-state model. If not, what is the nature of the deviation, and could further generalizations resolve the problems?

5. Job-performance measures. At the heart of all these questions is the issue of sensitivity of job performance to various training alternatives. As yet, no data are available that relate the effectiveness of simulation-based maintenance training to on-the-job performance.

REFERENCES

Rigney, J. W., Cremer, R. H., Towne, D. M., & Bond, N. A. Measurement and prediction of cognitive loadings in corrective maintenance tasks (Technical Report No. 52). Los Angeles: University of Southern California, Behavioral Technology Laboratories, April 1967.

Rigney, J. W., & Towne, D. M. MATGEN, a technique for generating symptom-malfunction matrices for complex devices. Los Angeles: University of Southern California, Behavioral Technology Laboratories, 1972.

Rigney, J. W., & Towne, D. M. Computer-aided performance training for diagnostic and procedural tasks. Journal of Educational Technology Systems, 1974, 2, 279-304.

Rigney, J. W., & Towne, D. M. Some concepts, devices, and procedures for improving technical training and on-the-job performance of maintenance (NAVTRAEQUIPCEN Technical Report 76-C-0023-1). Los Angeles: University of Southern California, Behavioral Technology Laboratories, August 1977.

Rigney, J. W., Towne, D. M., King, C. A., & Moran, P. J. Field evaluation of the generalized maintenance trainer-simulator: I. Fleet communications system (Technical Report No. 89). Los Angeles: University of Southern California, Behavioral Technology Laboratories, October 1978.

Rigney, J. W., Towne, D. M., Moran, P. J., & Mishler, R. A. Field evaluation of the generalized maintenance trainer simulator: II. AN/SPA-66 radar repeater (NPRDC Tech. Rep. 80-30). San Diego: Navy Personnel Research and Development Center, July 1980. (AD-A083 977)

Towne, D. M. ARMAN - A computer system for applying MTM (Methods Time Measurement), Proceedings, Conference of MTM Applications. Fairlawn, NJ: MTM Association for Standards and Research, 1968.

Towne, D. M., & Mason, A. K. Toward synthetic methods analysis. Journal of Industrial Engineering, 1967, 18, 52-56.

Wylie, C. D., & Bailey, G. V. Electronic equipment maintenance training system: Preliminary design options (NPRDC Tech. Note 79-3). San Diego: Navy Personnel Research and Development Center, October 1978.

APPENDIX

SUMMARY COMPUTER PROGRAM DOCUMENTATION

SUMMARY COMPUTER PROGRAM DOCUMENTATION

Name of Program: TRAINER

Purpose: This program drives a prototype computer-based training system known as the Generalized Maintenance Trainer Simulator (GMTS) or Rigney Trainer, a device that is currently used to simulate electronic equipment maintenance. Simulation is achieved by displaying one of a number of photographs of the equipment with a microfiche projector; interface with the user occurs via the projector, a sonic pen, and the CRT. A data base file contains a complete description of the equipment, simulated faults in the equipment, and a listing of the photographs, thus making TRAINER independent of the equipment that it simulates.

Hardware requirements: UCSD Pascal (TM) operating system version I.5, Bruning model 95 microfiche projector, Science Accessories Corporation Grafpen, and a Terak 8510 computer and 8512 disk drive. The source code has also been compiled under versions II.0 and II.1 of UCSD Pascal.

Documentation: Fully documented.

Comments: Four other programs related to the trainer system are available:

1. XY--Computes coordinates of touch points on the trainer panel.

2. PREPROCESS--Processes the alphanumeric data base (entered using the Pascal editor) into the format required by TRAINER.

3. UTILITY--General utility routines for disk initialization, etc.

4. DATACK--Data base checkout/verification program.

****

Name of Program: XY

Purpose: This program is to be used in conjunction with the TRAINER program to compute the coordinates of legitimate sonic pen touch points on the front panel of the trainer simulator. New touch points may be defined or old point positions revised; the results are stored in the file RTXYDATA.DATA for use by TRAINER.

Hardware Requirements: Same as for TRAINER.

Documentation: Fully documented.

****

Name of Program: PREPROCESS.

Purpose: This program is to be used in conjunction with the TRAINER program to process the alphanumeric data base. It processes a data base created using the Pascal long-text editor (L2) to create the file RTDATA.DATA, which is used by TRAINER.

Hardware Requirements: UCSD Pascal (TM) operating system.

Documentation: Fully documented.

****

Name of Program: UTILITY

Purpose: This program is to be used in conjunction with the TRAINER program to provide necessary utility routines for GMTS. Such routines include disk initialization, problem sequence selection, current sequence examination, student performance printouts, and a master copy routine.

Hardware Requirements: UCSD Pascal (TM) operating system.

Documentation: Fully documented.

****

Name of Program: DATACK

Purpose: This program is to be used in conjunction with the TRAINER program to verify the logical correctness of the data base used in the GMTS. It directs to the screen data that TRAINER would ordinarily send to the microfiche projector and reads its input from the keyboard rather than from a sonic pen. This allows a user who is familiar with the data base to trace errors made in preparing the data.

Hardware Requirements: UCSD Pascal (TM) operating system, version I.5, Terak 8510 computer with an 8512 disk drive, and a Science Accessories Corporation sonic Grafpen.

Documentation: Fully documented.